Node-attribute graph layout for small-world networks

Helen Gibson, Joe Faith
School of Computing, Engineering and Information Sciences
Northumbria University
Newcastle-upon-Tyne, UK
{*helen.gibson,joe.faith*}*@northumbria.ac.uk*

***Abstract*—Small-world networks are a very commonly occurring type of graph in the real-world, which exhibit a clustered structure that is not well represented by current graph layout algorithms. In many cases we also have information about the nodes in such graphs, which are typically depicted on the graph as node colour, shape or size. Here we demonstrate that these attributes can instead be used to layout the graph in high-dimensional data space. Then using a dimension reduction technique, targeted projection pursuit, the graph layout can be optimised for displaying clustering. The technique outperforms force-directed layout methods in cluster separation when applied to a sample, artificially generated, small-world network.**

***Keywords*-Graph Layout; Dimension Reduction; Node-attribute; Clustering; Small-world**

I. Introduction

Many real-world networks display a small-world network structure, characterised by the fact that they are highly clustered and have smaller than average shortest path lengths. Small-world networks are likely to contain cliques (a fully connected subgraph) or at least highly connected subgraphs of nodes. Real world examples include neural networks, social networks and the connectivity of the World Wide Web [1]. It has been found that when users arrange such graphs manually, they will seek to organise the graph such that nodes in clusters are grouped together [2]. It would therefore be useful for graph layout algorithms to also emphasise these clusters – but most layout algorithms fail to do this.

Node-attribute graphs are graphs in which all nodes have a set of attributes associated with them. These attributes can be thought of as a new type of node to which that nodes links or, vice-versa, that other nodes in the graph could instead be defined as attributes. For example, membership of a group in a social network could be represented as an attribute of those nodes representing its members, or as a link from a group node to the member nodes. Here we demonstrate that attributes associated with cluster membership can position each node in a high-dimensional attribute space such that dimension reduction techniques can then be used to layout the nodes in the graph in two dimensions. The aim of this technique is first to show the clustering in the graph and ultimately to use this information to analyse which attributes are most influential in the clustering and the layout in general.

This pilot study uses a dimension reduction technique developed for vector data, targeted projection pursuit, to show cluster structure more clearly than other layout algorithms.

II. Small-World Networks

Many real-world networks can be approximated by small-world networks. In fact, Albert and Barabási [3] have hypothesised that the prevalence of small-world networks in biological systems is due to inherent structural advantages. A small-world network is where, despite the fact that the network is large, it takes very few steps to move between any two nodes. Specifically, they have a smaller than average shortest path length and a high clustering coefficient meaning they are also more likely to contain clusters of nodes. The most common real-world example of a small-world network is from the six degrees of separation experiment; the concept that most people in the United States are separated by only six people in a chain of friendship, as suggested by psychologist Milgram [4]. Other examples of small-world networks include the collaboration of actors in films [5], social networks, neural networks of the brain [6], and the connectivity of the World Wide Web [7].

Given that small-world networks are such a commonly occurring graph structure, it is then a surprise that so few layout algorithms display them well [8]. Force-directed layouts, in particular, do not optimise the visualisation for small world networks. This is, in part, because of the short path length (graph-theoretic distance) small-world networks have. Force-directed layouts such as Kamada and Kawai's [9] energy-based layout try to represent graph-theoretic distance as Euclidean distances and so if all pairs of nodes have a small graph-theoretic distance then most pairs of nodes are placed close together and the clustered structure of the graph is lost. Therefore layouts which can accentuate this clustered structure offer advantages over traditional layouts for small-world networks.

III. Node Attribute Graphs

Node-attribute, or multivariate, graphs are graphs that incorporate attributes on the nodes as well as displaying the links between the nodes [10]. Node attributes on graphs are quite common and the ability to represent them by colour, shape or size is a functionality included in many pieces

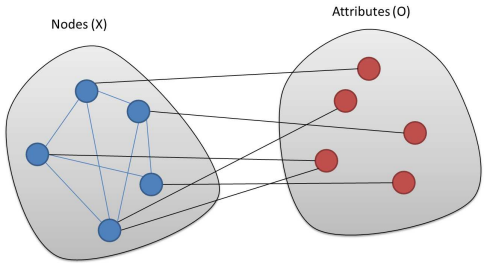


Figure 1: A node-attribute graph where the blue circles are nodes and the red are attributes. Links exist between the blue nodes and between the blue and red nodes.

of graph drawing software such as Cytoscape [11], Gephi [12], Pajek [13] and others. However, there is a limit to the number of attributes that can be represented this way. One way of representing a node attribute in a graph would be to add an extra node representing the attribute and a (weighted) link to the graph from the node whose attribute it is, as in Fig. 1. Obviously actually representing the graph this way would add a significant number of extra nodes and links to the view and most likely make the graph harder, not easier, to read.

Instead, defining the graph this way means the graph can be divided into two separate graphs: the original graph with no attributes shown and a bipartite graph where links only exist between nodes and their attributes. This type of graph structure contains a subset of graphs termed semi-bipartite graphs [14] where a semi-bipartite graph has second type of nodes as opposed to a set of attributes. Real-world graphs which have this semi-bipartite structure include Xu et al.'s [14] network containing genes and gene ontology terms where genes are connected to their ontology terms and the terms are linked to each other hierarchically. Other possible real-world examples could be a drug and protein network where similar drugs (or similar proteins) are linked and a drugs are linked to proteins they target [15]. Similarly in social networks, such as those from Facebook, links are made between friends and a second set of links can be added for connections to groups, activities, 'Likes', 'Fan of', etc.

Another example of multi-modal graphs are those from formal concept analysis, known as Galois or concept lattices. These are similar to bipartite graphs but for which a specific graph visualisation has been developed. The set of nodes are divided into non-disjoint subsets each of which contains nodes that share the same attributes; and the relations between subsets are then shown using a Hasse diagram [16]. The composition of each subset is then shown using by annotating the glyph representing it. Freeman and White [17] used Galois lattices to show social networks with three types of link: node-node, attribute-attribute and node-attribute. However they are different from the graphs we are visualising here as only node-attribute data is used and then the visualisation is used to imply the node-node and attribute-attribute relationships rather than taking them as a given from the start.

IV. Graph Clustering

Users value clustering in graphs and they try to recreate this structure when laying out graphs manually [2]. Traditional force-directed layouts do not reproduce the clusterings in graphs well; this is because they tend to place all nodes of high degree at the centre of the graph and also try to adhere to the aesthetic criteria of keeping edge lengths uniform which makes cluster separation more difficult [8]. One attempt to visualise clusterings in graphs is due to Noack [8], [18] who demonstrated an energy layout algorithm for clustering graphs, calling them 'interpretable layouts' since the links are not shown in the visualisation but are instead used to position the nodes; the nodes are then also sized depending on their degree. The graph is not clustered prior to layout, rather it is clustered based on the graph-partitioning idea of cuts and then visualised. A cut is a simple measure of the coupling between two sets of disjoint nodes, and Noack [18] proposes two models: node-repulsion and edge-repulsion. The node-normalised cut is the ratio of number of edges between the two partitions to the total possible number of edges between the two partitions. The edge-normalised cut is then defined as the ratio between the number of edges between the partitions and the product of the sums of the degrees of the nodes in each partition. The edge-repulsion model is preferred as it is less likely to place nodes of high degree in the centre of the graph.

Other attempts for visualising clustered graphs include Huang and Nyguen's [19] approach where the graph is divided into densely connected subgraphs that are each placed on their own separate rectangular partition for layout. Chung Graham and Tsiatas [20] use a version of Kamada and Kawai's force-directed layout and the PageRank algorithm for computing a clustered layout while Balzer and Deussen [21] use a 3-D graph with pre-defined clusters to first wrap spheres around clusters and then use implicit surfaces to further emphasise cluster separation.

V. Dimension Reduction and Targeted Projection Pursuit

Dimension reduction takes some data in high-dimensional space and computes a lower dimensional representation of that data, which for visualisation purposes is likely to be two dimensions. Methods of dimension reduction include multidimensional scaling, principal component analysis and other linear and non-linear methods.

Targeted projection pursuit (TPP) [22] is a linear projection method of dimension reduction such that, instead of searching for the most interesting projection (as with projection pursuit), the user can interact with the data by attempting to move the points around to fit their intuition and

the algorithm will try to find a projection that best matches the users desired view. This is an effective technique because it allows users to explore and interact with the data in real-time as well as to iteratively make and test hypotheses about how the data can be projected and what that projection then means in the context of the original high-dimensional data set. TPP works by the user suggesting a view of the data they wish to see and then searches for a projection that best matches that target view. So by taking an $n \times k$ matrix X and a $n \times 2$ target view T TPP tries to find a $k \times 2$ projection matrix P that minimises the difference between the two, where n is the number of points and k the number of dimensions. That is

$$\min \|T - XP\| \tag{1}$$

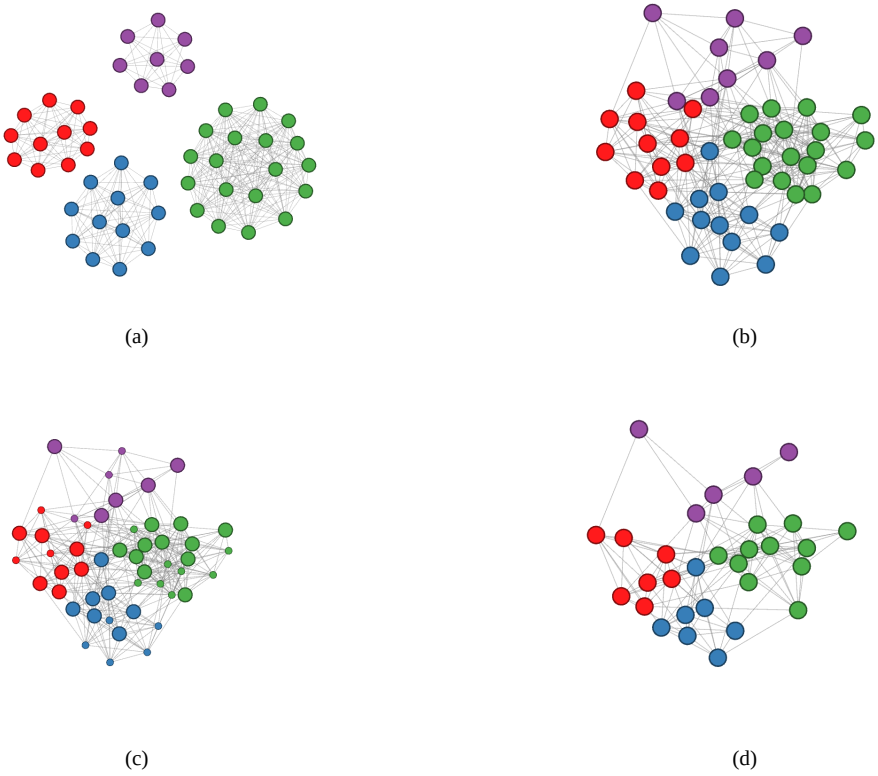
As an alternative to user-directed layout, TPP can also search for a projection that separates the data into pre-defined classes by trying to maximise the distance between classes through projecting the data on to the vertices of a simplex [23].

VI. Node-Attribute Graph Layout

Define a node-attribute graph to be $G(V_X, V_O, E_{XX}, E_{XO})$ where V_X are the nodes in first partition, V_O are the nodes in the second partition, E_{XO} are the edges linking nodes in V_X to V_O and E_{XX} are the edges between the nodes in partition V_X. We call the nodes in V_X our entity nodes and the nodes in V_O our attribute nodes.

One pre-requisite for this visualisation is that each node needs to be defined as a being a member of a particular cluster before the analysis can be carried out. This can be done by using particular cluster structure that occurs naturally in the dataset or by using an unsupervised clustering algorithm first, such as k-means, to impose a clustered structure on the dataset.

The visualisation of the graph will show the V_X nodes and the E_{XX} edges while the layout will depend on the clustering of the nodes and the edges E_{XO} between the V_X nodes, that are visualised, and the V_O nodes, that are not. In order to layout the points, for each node in V_X a vector, p_i with binary entries is constructed based on their links to the V_O nodes, i.e. if an edge between V_{X_i} and V_{O_j} exists



(a) (b) (c) (d)

Figure 2: Dataset creation. (a) The four initial cliques of nodes. (b) Random links are added between cliques and removed within cliques. (c) Nodes that are chosen as attributes are indicated as the smaller nodes. (d) The attribute nodes and their connections are removed leaving only the $x - x$ links. All layouts produced using Yifan Hu's force directed layout in Gephi.

then the entry takes value one and if it does not then takes value zero.

$$p_i = \{(p_{i1}, p_{i2}, p_{i3}, \ldots)^T \mid p_{ij} = 1 \text{ if } E_{X_i,O_j} \text{ exists and } 0 \text{ otherwise}\} \quad (2)$$

This can be extended to include relationships to attributes which are not only binary but also nominal and real-valued data too, especially if the link is considered to have a weighting. From this each V_X can be described as $V_X\,(c, p_i)$ in $|V_O + 1|$ dimensional space where one of those dimensions describes the cluster, c, to which the node belongs.

The aim is then to use TPP to visualise the position of each node in two-dimensional space and use it to separate the clusters in the graph as far as possible. In this case the vector p_i for each of the V_{X_i} nodes is taken as one of the n rows forming the $n \times k$ matrix, where k is the number of attribute nodes. A two-dimensional projection is found that minimises the difference between the target projection defined by the user and itself. The nodes can be coloured according to their cluster membership or if no clustering is proposed then an unsupervised clustering algorithm can be used to define one. The links between the entity nodes can then be added to the visualisation.

From this point the user can then either repeatedly select and drag nodes to move them to fit their idea of how the graph should appear and the closest possible projection will be shown or the process can be automated. In this case to have the centre of each cluster to be positioned over the vertices of a simplex is seen as the optimum target view, i.e. where each of the clusters will be most separated from each of the other clusters. This automated process is akin to just the user trying to separate the clusters themselves by dragging points but achieves maximum separation.

VII. Example Application

In this pilot study, TPP was used to visualise a clustered small-world network with node-attribute data, and the results compared with the same graph visualised using the Yifan Hu layout in Gephi [12] and Noack's LinLog layout [18].

An example graph with the required properties (small-world, known clusters, and node attributes) was constructed by starting with several fully-connected cliques that will define the clusters in the graph (Fig. 2a). Specifically, using an artificially generated data set allows control over the properties of the graph in order to evaluate the potential success of the technique for real-world data in the future without having to account for noise or unexpected variations. The adjacency matrix that defines the graph was then randomly mutated to add new links between cliques and removing some links within cliques (Fig. 2b). Nodes were then randomly divided into entities and attributes (Fig. 2c) and any remaining links between two attributes are removed. From this there is data for two graphs: the bipartite graph between attribute and entity nodes and the graph of connections between the entity nodes only (Fig. 2d).

In this case the graph original consisted of 50 nodes with 318 links divided unequally into 4 cliques of sizes 11, 12, 19 and 8 and the addition of noise to the dataset increased this to 350 edges. Then the nodes were split into entity and attribute node groups with 30 nodes in the entity group and 20 nodes in the attribute group. This resulted in cluster sizes of 7, 7, 11 and 5 in the entity group and 4, 5, 8 and 3 in the attribute group. In terms of links this gives 132 links in the visualisation, 173 links used in the projection and 45 links between the attributes were removed. The graph is then laid out in three ways: the Yifan Hu force-directed approach from Gephi [12] (Fig. 3); TPP (Fig. 4); and Noack's LinLog layout in Fig. 5.

TPP clearly achieves a greater visual separation between clusters than the force-directed layout. This is especially the case with some nodes which would be difficult to determine which cluster they belong to without colouring. This could be advantageous in the future as it would free the use of colour to show some other attribute. The use of TPP to separate the clusters is different to just using user choice to position the nodes, as in Fig. 3, since in that case the position of the nodes is purely dependent on where the user want to put them. In TPP, however, the position of the nodes is the product of a linear projection. Additionally moving one node, or a group of nodes, in TPP rarely affects only the chosen nodes; other nodes are moved as a consequence of trying to fit the selected nodes to their preferred position. That is, the position of the nodes using the TPP algorithm is purely dependent on the attribute nodes and to a lesser extent cluster membership.

LinLog also creates clear spatial separation; however it imposes its own clustering on the data which makes clear comparisons difficult. While the lack of links in this layout makes the clustering very clear – and the distances between clusters gives an indication of the number of links between them – the lack of links means some of the understanding of how the clusters are related to each other is lost. It also affects the ability to see if there are individual links between nodes in different clusters that show interesting information.

VIII. Conclusion

The aim of this approach is to show the clustering that occurs in most small-world networks and its relationship to node attributes. It can be seen that the layout produced by TPP does show the clustered structure of the graph more clearly than a simple force-directed layout did where the separation of clusters is mostly discernible by their colour.

Further validation on this layout and its success will include measuring both the intra-node distances within clusters and the inter-node distances between clusters and comparing them between the layouts. Secondly, as it is known that users also prefer fewer edge crossings [2] in their graphs,

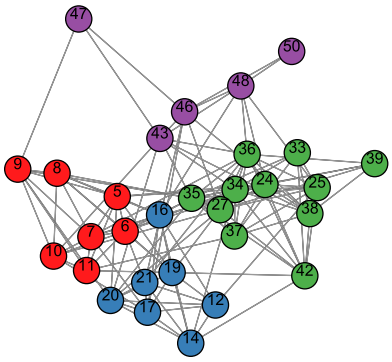


Figure 3: Yifan Hu's force-directed approach from Gephi

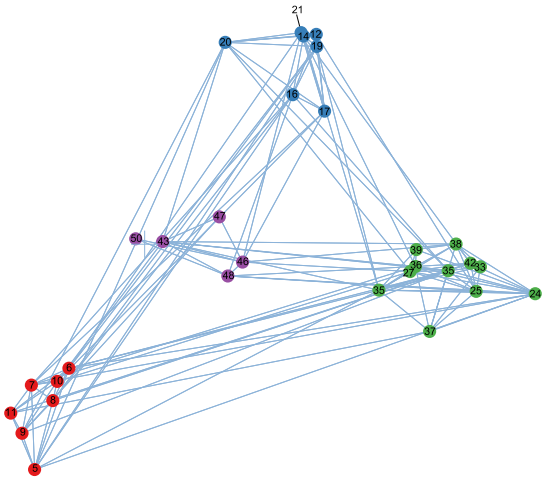


Figure 4: Target Projection Pursuit with clusters separated as far as possible

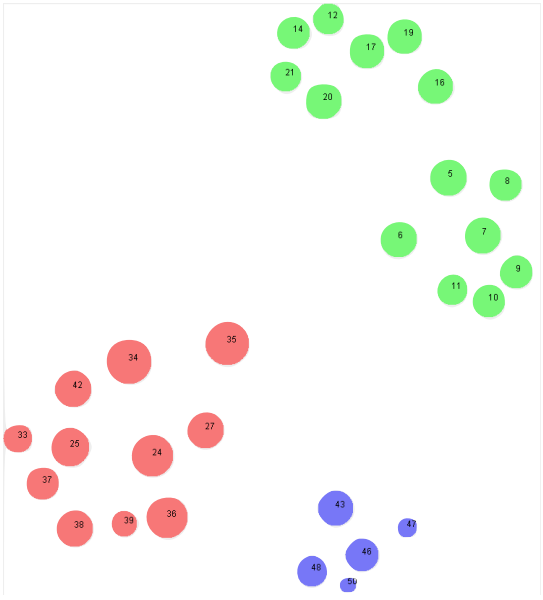


Figure 5: Noack's linlog layout which imposes its own clustering on the graph. The blue cluster corresponds to the purple cluster, the red to the green and the two green clusters to the red and blue clusters in the other layouts.

it will be useful to measure the difference between the numbers of edge crossings between the two layouts that show links. The purpose of using attributes for layout is not only important for producing a good layout, they may also be able to give more insight into the structure of the graph. Particularly, being able to assess which attributes may be the most influential in the layout and which attributes are the least, or even completely irrelevant.

Further extensions to this dataset would be to alter the ratio of entities to attributes and measure how this affects the ability to cluster the data and the layout in general. This was also an artificially created dataset and so most real-world graphs may contain more noise, specifically it would be useful to investigate how introducing known exceptions into the data, such as misleading attributes and wrongly classified nodes may give an indication of how great an effect they have on the layout and how easy is it to identify any errors. It will also be important to test how this technique can scale to graphs with hundreds or even thousands of nodes.

REFERENCES

[1] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[2] F. van Ham and B. Rogowitz, "Perceptual Organization in User-Generated Graph Layouts," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1333–1339, Nov. 2008.

[3] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, Jan. 2002.

[4] S. Milgram, "The small world problem," *Psychology Today*, vol. 2, no. 1, pp. 60–67, 1967.

[5] D. Auber, Y. Chiricota, F. Jourdan, and G. Melancon, "Multiscale visualization of small world networks," *Information Visualization, IEEE Symposium on*, p. 10, 2003.

[6] O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag, "Organization, development and function of complex brain networks." *Trends in cognitive sciences*, vol. 8, no. 9, pp. 418–25, Sep. 2004.

[7] R. Albert, H. Jeong, and A. Barabási, "Internet: Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, Sep. 1999.

[8] A. Noack, "An energy model for visual graph clustering," in *Graph Drawing*, ser. Lecture Notes in Computer Science, G. Liotta, Ed. Springer Berlin / Heidelberg, 2004, vol. 2912, pp. 425–436.

[9] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, vol. 31, no. 1, pp. 7–15, 1989.

[10] M. Wattenberg, "Visual exploration of multivariate graphs," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 2006, pp. 811–819.

[11] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome Research*, vol. 13, no. 11, pp. 2498–504, Nov. 2003.

[12] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," in *Third International AAAI Conference on Weblogs and Social Media*, 2009, pp. 361–362.

[13] V. Batagelj and A. Mrvar, "Pajek analysis and visualization of large networks," in *Graph Drawing*, ser. Lecture Notes in Computer Science, P. Mutzel, M. Jnger, and S. Leipert, Eds. Springer Berlin / Heidelberg, 2002, vol. 2265, pp. 8–11.

[14] K. Xu, R. Williams, S.-H. Hong, Q. Liu, and J. Zhang, "Semi-bipartite graph visualization for gene ontology networks," ser. Lecture Notes in Computer Science, D. Eppstein and E. R. Gansner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 5849, pp. 244–255–255.

[15] S. J. Cockell, J. Weile, P. Lord, C. Wipat, D. Andriychenko, M. Pocock, D. Wilkinson, M. Young, and A. Wipat, "An Integrated Dataset for in Silico Drug Discovery," *Journal of Integrative Bioinformatics*, vol. 7, no. 3, pp. 116–128, Jan. 2010. [Online]. Available: http://journal.imbio.de/index.php?paper_id=116

[16] U. Priss, "Formal concept analysis in information science," *Annual Review of Information Science and Technology*, vol. 40, no. 1, pp. 521–543, 2006.

[17] L. C. Freeman and D. R. White, "Using Galois Lattices to Represent Network Data," *Sociological Methodology*, vol. 23, pp. 127–146, 1993.

[18] A. Noack, "Energy models for graph clustering," *Journal of Graph Algorithms and Applications*, vol. 11, no. 2, pp. 453–480, 2007.

[19] M. L. Huang and Q. V. Nguyen, "A fast algorithm for balanced graph clustering," in *Information Visualization, 2007. 11th International Conference*, July 2007, pp. 46 –52.

[20] F. Graham and A. Tsiatas, "Finding and Visualizing Graph Clusters Using PageRank Optimization," in *Algorithms and Models for the Web-Graph*, ser. Lecture Notes in Computer Science, R. Kumar and D. Sivakumar, Eds. Springer Berlin Heidelberg, 2010, vol. 6516, pp. 86–97.

[21] M. Balzer and O. Deussen, "Level-of-detail visualization of clustered graph layouts," *Asia-Pacific Symposium on Visualization*, pp. 133–140, 2007.

[22] J. Faith, "Targeted projection pursuit for interactive exploration of high- dimensional data sets," in *11th International Conference Information Visualization*, 2007, pp. 286–292.

[23] J. Faith, R. Mintram, and M. Angelova, "Targeted projection pursuit for visualizing gene expression data classifications," *Bioinformatics*, vol. 22, no. 21, pp. 2667–2673, Nov. 2006.